PROFESSIONAL GUIDE

DevOps Support Services

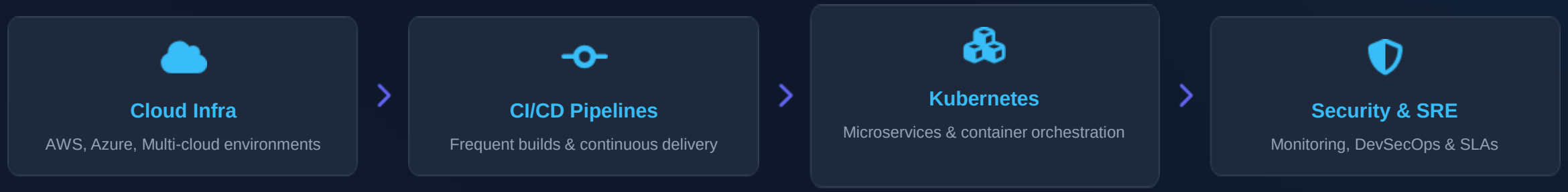
A Practical Guide to Cloud, CI/CD, Kubernetes, Security, SRE and MLOps

Sustaining mission-critical operations, release speed, infrastructure security, and system reliability.

Enterprise Infrastructure & Operations Presentation Series

Why DevOps Support Matters

Modern applications become exponentially more complex as engineering teams scale operations, expand services, and accelerate deployment frequency.



Core Challenge: As operational surface area expands, maintaining 99.9%+ availability requires dedicated expertise beyond software development.

What Are DevOps Support Services?

Infrastructure Support

Maintaining cloud resources, network topologies, storage, and server configurations.

CI/CD Pipeline Support

Managing automated build, test, and release deployment pipelines for dev teams.

Infrastructure as Code

Managing Terraform, CloudFormation, and Ansible templates without configuration drift.

Monitoring & Alerting

Configuring metrics, logs, synthetic checks, and real-time incident notifications.

Release Management

Ensuring zero-downtime blue/green or canary deployments across staging and production.

Production Troubleshooting

Diagnosing performance bottlenecks, container failures, and infrastructure errors.

⚖️ Implementation vs. Ongoing Support

Focus Area	DevOps Implementation	Ongoing DevOps Support
CI/CD Pipelines	Build pipelines & setup automation	Maintain, optimize & debug build failures
Infrastructure	Design & provision initial cloud setups	Operate, patch, scale & optimize costs
Observability	Configure metrics & alerting tools	Monitor 24/7, triage alerts & fine-tune triggers
Deployments	Establish deployment strategies	Troubleshoot failed releases & rollback errors
Automation	Write IaC scripts & playbooks	Maintain playbooks & resolve configuration drift

Key Distinction: Implementation establishes the DevOps foundations, while Ongoing Support ensures long-term system health, resilience, and evolution post-launch.

Key Areas of DevOps Support

Cloud Operations

Multi-cloud governance, IAM management, VPC peering, and storage lifecycle.

Container Ecosystem

Kubernetes cluster lifecycle, Docker containerization, Helm releases, and registry maintenance.

DevSecOps & Policy

Vulnerability scanning, secrets rotation, policy enforcement, and compliance checks.

SRE & Incident Ops

SLI/SLO tracking, error budget management, on-call rotation, and root cause analysis.

IaC & Config Management

Maintaining Terraform modules, Ansible playbooks, and GitOps workflows.

Observability Stack

Prometheus, Grafana, ELK/OpenSearch, Datadog, and OpenTelemetry instrumentation.

MLOps Infrastructure

Model serving pipelines, GPU cluster setup, feature store support, and data pipelines.

Managed DevOps Services

Core Operational Activities

- **Continuous Pipeline Maintenance:** Updating plugins, security patches, and build scripts.
- **Cloud Cost Optimization:** Identifying idle resources, rightsizing instances, and managing savings plans.
- **Automated Backups & DR:** Testing disaster recovery procedures and snapshot policies regularly.
- **Security Patching:** Applying OS-level patches and dependency updates without downtime.

Strategic Value Deliverables

- **Proactive Capacity Planning:** Anticipating traffic spikes and configuring autoscaling groups.
- **Release Management Support:** Assisting dev teams during complex production rollouts.
- **Runbook Documentation:** Creating and maintaining standard operational procedures (SOPs).
- **Infrastructure Audits:** Conducting periodic architectural reviews against cloud best practices.

24/7 DevOps Support Services

When Continuous 24/7 Support Is Essential

- **Global Customer Platforms:** Products serving users across multiple international time zones.
- **Mission-Critical Workloads:** Financial, healthcare, or SaaS systems with zero-downtime SLAs.
- **High-Frequency Deployments:** Teams executing continuous releases around the clock.
- **E-Commerce & High Traffic:** Platforms where downtime directly incurs heavy revenue losses.

Support Coverage Considerations

Not every organization requires 24/7 shift-based engineering support.

Smaller platforms or internal tooling environments often thrive on a hybrid model: **business-hours active support combined with automated high-severity incident alerts**.

Kubernetes Support Services

Kubernetes orchestration simplifies application scaling but introduces significant operational complexity across cluster management.

Cluster Administration

Worker node provisioning, control plane monitoring, version upgrades, and etcd backups.

Workload Management

Deployments, StatefulSets, DaemonSets, HPA/VPA scaling, and pod resource request tuning.

Networking & Ingress

CNI plugin maintenance, Ingress controllers, TLS cert management, and Service Mesh support.

RBAC & Security

Cluster roles, ServiceAccounts, NetworkPolicies, and secret management integrations.

Persistent Storage

StorageClasses, CSI drivers, persistent volume provisioning, and backup validation.

Cluster Troubleshooting

Resolving CrashLoopBackOff, OOMKilled errors, node pressure, and pending pods.

Kubernetes Troubleshooting Example

Scenario: Production Pods Keep Restarting (CrashLoopBackOff)

1. Pod Status
`kubectl get pods`

→

2. Inspect Events
`kubectl describe`

→

3. Container Logs
`kubectl logs --previous`

→

4. Health Probes
Check liveness/readiness

5. Resource Limits
Check for OOMKilled

→

6. Node Health
Check disk/memory pressure

→

7. Configuration
Verify ConfigMaps/Secrets

→

8. Remediation
Apply fix & verify health

Systematic investigation ensures rapid mean time to resolution (MTTR) while preventing recurring cluster outages.

AWS DevOps Support Services

Core AWS Managed Services

- **Compute & Container Services:** Managing Amazon EC2, Elastic Kubernetes Service (EKS), and Elastic Container Service (ECS).
- **Serverless Architecture:** Supporting AWS Lambda, API Gateway, and EventBridge workflows.
- **Identity & Governance:** Managing IAM policies, AWS Organizations, SCPs, and CloudTrail audits.
- **Networking & Delivery:** VPC peering, Transit Gateways, Route 53, and CloudFront CDN management.

Automation & Observability

- **Infrastructure as Code:** Provisioning via Terraform and AWS CloudFormation templates.
- **AWS Native CI/CD:** Managing CodePipeline, CodeBuild, and CodeDeploy workflows.
- **CloudWatch & X-Ray:** Monitoring metrics, alarms, log groups, and distributed traces.
- **Scope Beyond SysAdmin:** Broad cloud automation and architecture optimization rather than basic server administration.

Azure DevOps Support Services

Azure Pipeline & Ecosystem

- **Azure Pipelines:** Managing YAML pipeline definitions, self-hosted build agents, and release gates.
- **Azure Kubernetes Service (AKS):** Operating AKS clusters, node pools, and Azure CNI integration.
- **Azure Infrastructure Automation:** Deploying Bicep templates and ARM templates cleanly.
- **Azure Artifacts & Repos:** Managing feeds, package versioning, and branch policies.

Security & Operations

- **Identity Management:** Integrating Microsoft Entra ID (Azure AD) and Managed Identities.
- **Azure Monitor & Log Analytics:** Querying KQL logs, setting up alerts, and Application Insights.
- **Azure Key Vault:** Securing connection strings, TLS certificates, and API keys.
- **Production Support:** Continuous troubleshooting for enterprise hybrid-cloud workloads.

DevSecOps Support Services

Integrating security testing seamlessly throughout the software development lifecycle to catch vulnerabilities early.

Code
SAST & Secret checks

Build
Dependency scanning

Test
DAST & API security

Scan
Container image audit

Deploy
IaC policy enforcement

Monitor
Runtime protection

Automation & Scanning

Integrating SonarQube, Trivy, Snyk, and OWASP ZAP into build pipelines to automate compliance checks.

Secrets & Governance

Enforcing HashiCorp Vault integration, OPA (Open Policy Agent) rules, and CIS benchmark hardening.

SRE Support Services

Core SRE Principles

- **SLIs & SLOs:** Defining Service Level Indicators and establishing realistic Service Level Objectives.
- **Error Budgeting:** Balancing release velocity against system reliability requirements.
- **Toil Reduction:** Replacing repetitive operational tasks with automated software scripts.
- **Post-Incident Reviews:** Conducting blameless post-mortems to drive continuous improvement.

SLO Practical Example

Target: 99.9% Uptime SLO over 30 days.

Error Budget: Allows 43.2 minutes of total downtime per month. If feature releases consume 35 minutes early in the month, releases slow down to protect reliability.

MLOps Support Services

Machine learning workloads require distinct production operational practices beyond traditional web application pipelines.

ML Pipelines

Automating data extraction, feature engineering, and model training pipelines (Kubeflow, Airflow).

Model Serving

Deploying models via REST/gRPC endpoints using Triton, MLflow, or AWS SageMaker.

Drift Monitoring

Detecting data drift, concept drift, and performance degradation in production models.

GPU Infrastructure

Managing GPU node pools, resource allocation, and optimizing cloud compute overhead.

Feature Store Support

Maintaining consistent offline and online feature stores for real-time inference.

Automated Retraining

Triggering model retraining workflows automatically upon performance threshold breach.

Observability in DevOps Support

Observability provides deep visibility into complex distributed systems, going beyond basic uptime ping checks.

Metrics
Numeric values (CPU, latency, RPS)

+

Logs
Timestamped event records

+

Traces
End-to-end request journeys

=

Observability
Full System Telemetry

Faster Investigation
Pinpoint exact microservice failures in seconds

→

Better System Reliability
Reduced MTTR & proactive threat mitigation

DevOps Incident Response Workflow

1. Detection
Alert triggers via PagerDuty/Slack

→

2. Triage
Verify incident & assign lead

→

3. Impact
Assess blast radius & affected users

→

4. Investigation
Analyze metrics, logs & telemetry

→

5. RCA
Determine underlying root cause

6. Mitigation
Apply rollback, patch, or traffic reroute

→

7. Recovery
Restore full service functionality

→

8. Validation
Confirm system stability with tests

→

9. Documentation
Record timelines & action items

→

10. Prevention
Implement long-term fixes

⚠ Common DevOps Challenges

- **1. CI/CD Pipeline Failures:** Flaky tests, broken dependencies, and slow build queues.
- **2. Kubernetes Failures:** CrashLoopBackOff, node pressure, and ingress routing issues.
- **3. Configuration Drift:** Discrepancies between staging and production environments.
- **4. Cloud Resource Waste:** Unused provisioned resources inflating monthly cloud bills.
- **5. Alert Fatigue:** Excessive non-actionable alerts overwhelming on-call engineers.
- **6. Monitoring Gaps:** Lack of visibility into distributed microservice interactions.
- **7. Security Vulnerabilities:** Outdated container images and unpatched OS dependencies.
- **8. Deployment Outages:** Unplanned downtime during complex production releases.
- **9. Capacity Bottlenecks:** Sudden traffic spikes crashing unscaled database layers.
- **10. Backup Failures:** Untested disaster recovery procedures causing data loss.

How Automation Helps

Automation replaces manual, error-prone operations with repeatable, testable software scripts.

IaC Provisioning

Eliminating manual cloud console changes with audited Terraform modules.

Automated Rollbacks

Instantly reverting failed deployments based on synthetic test triggers.

Smart Alert Routing

Filtering noise and routing critical events directly to responsible on-call leads.

Automated Patching

Running scheduled security scans and automated pull requests for image upgrades.

Best Practice: All production automation must be tested in staging and controlled through peer-reviewed GitOps workflows.

Internal vs Managed Support Models

Factor	In-House DevOps Team	Managed DevOps Support	Hybrid Support Model
Operational Control	High internal control	Shared service model	Shared (In-house leads, extended support)
Domain Expertise	Limited to hired staff skillsets	Access to specialized multi-domain engineers	Combined internal context & specialist depth
Support Availability	Subject to team bandwidth & turnover	Defined SLA coverage commitments	Flexible coverage for peak hours & on-call
Cost Structure	Fixed salaries, benefits & training overhead	Predictable service engagement pricing	Mixed operational expense structure
Scalability	Constrained by hiring lead times	Rapidly scalable up or down	Flexible expansion for special projects

Evaluation Checklist

Key criteria to evaluate when selecting an external DevOps support partner:

- Multi-Cloud Expertise (AWS, Azure, GCP)
- Infrastructure as Code & GitOps Mastery
- DevSecOps & Security Compliance Practices
- Clear SLA & Incident Response Escalations
- Kubernetes & Container Orchestration Depth
- Observability Stack Management Capabilities
- Proven Incident Response & Post-Mortem Workflows
- Comprehensive Documentation & Knowledge Transfer

Global Delivery & India Operations

Key Operational Advantages

- **Extensive Engineering Talent:** Deep talent pool across cloud native, Kubernetes, and SRE disciplines.
- **Round-the-Clock Timezone Coverage:** Natural alignment for 24/7 follow-the-sun operations across US, Europe, and Asia.
- **Flexible Engagement Models:** Ability to augment internal engineering teams or provide fully managed services.
- **Cost-Effective Operations:** High-value engineering support delivered without compromising operational quality.

Strategic Remote Integration

Modern DevOps support companies operate as seamless extensions of client engineering teams using shared Slack/Teams channels, Jira queues, and transparent Git repositories.

Operational Readiness Checklist

- [x] CI/CD Pipelines documented & versioned
- [x] Centralized log aggregation configured
- [x] Infrastructure tracked in version control
- [x] Automated database backups tested
- [x] IAM roles & cloud permissions audited
- [x] Security vulnerability scanning enabled
- [x] Production metrics & dashboards active
- [x] High-severity alerts & thresholds defined
- [x] IaC modules validated without drift
- [x] Kubernetes cluster health checks active
- [x] Secrets stored securely (Vault/KeyVault)
- [x] Incident runbooks & escalations ready

🔑 Key Takeaways

- **Reliability First:** DevOps support turns static infrastructure setups into resilient, operational environments.
- **Automation Drives Speed:** Standardizing releases with IaC and CI/CD reduces human error and downtime.
- **Security Is Continuous:** DevSecOps integrates vulnerability checks into daily deployment flows.
- **Observability Matters:** Combining metrics, logs, and traces enables rapid MTTR during outages.
- **Kubernetes Needs Depth:** Container orchestration requires dedicated cluster-level operational care.
- **SRE Protects Velocity:** SLOs and error budgets strike the right balance between speed and stability.
- **Choose Flexible Models:** Select in-house, managed, or hybrid support based on internal capacity.

🏁 Conclusion

> DevOps support becomes increasingly vital as organizations operate complex cloud environments, rapid deployment schedules, Kubernetes clusters, security-sensitive applications, and mission-critical production workloads.

Organizations should select their DevOps support strategy based on:

Infra Complexity

Multi-cloud vs single provider setups.

Uptime SLA

24/7 availability vs business-hours needs.

Security Needs

Compliance & policy requirements.

Team Capacity

In-house bandwidth vs external expertise.